# DESIGN and VALIDATION of RELIABLE NETWORKED SYSTEMS

Professor R. K. Iyer[1]
Research Assistant Professor Z. Kalbarczyk[1], J. Patel[2]
System engineer D. Burke[1]
Graduate students:
L. Chen[1], M. S. Lee[1], N. Chazin[1], G. Ries[1], Y. Xiao[1]

[1]Center for Reliable and
High-Performance Computing
University of Illinois at Urbana-Champaign

[2]JPL - NASA Jet Propulsion Laboratory
Pasadena, California

# Overview

- **DEPEND hierarchical framework for design and evaluation of dependable systems**
  - the complete case study to demonstrate the capabilities of the hierarchical simulation methodology for accurate fault modeling
    - target system: Myrinet, high-speed network
- **Validation of the simulation results via actual fault injection to the same target system using N-FTAPE**
- **Collaboration with JPL**

# DEPEND: A Hierarchical Framework for Design of Dependable Systems

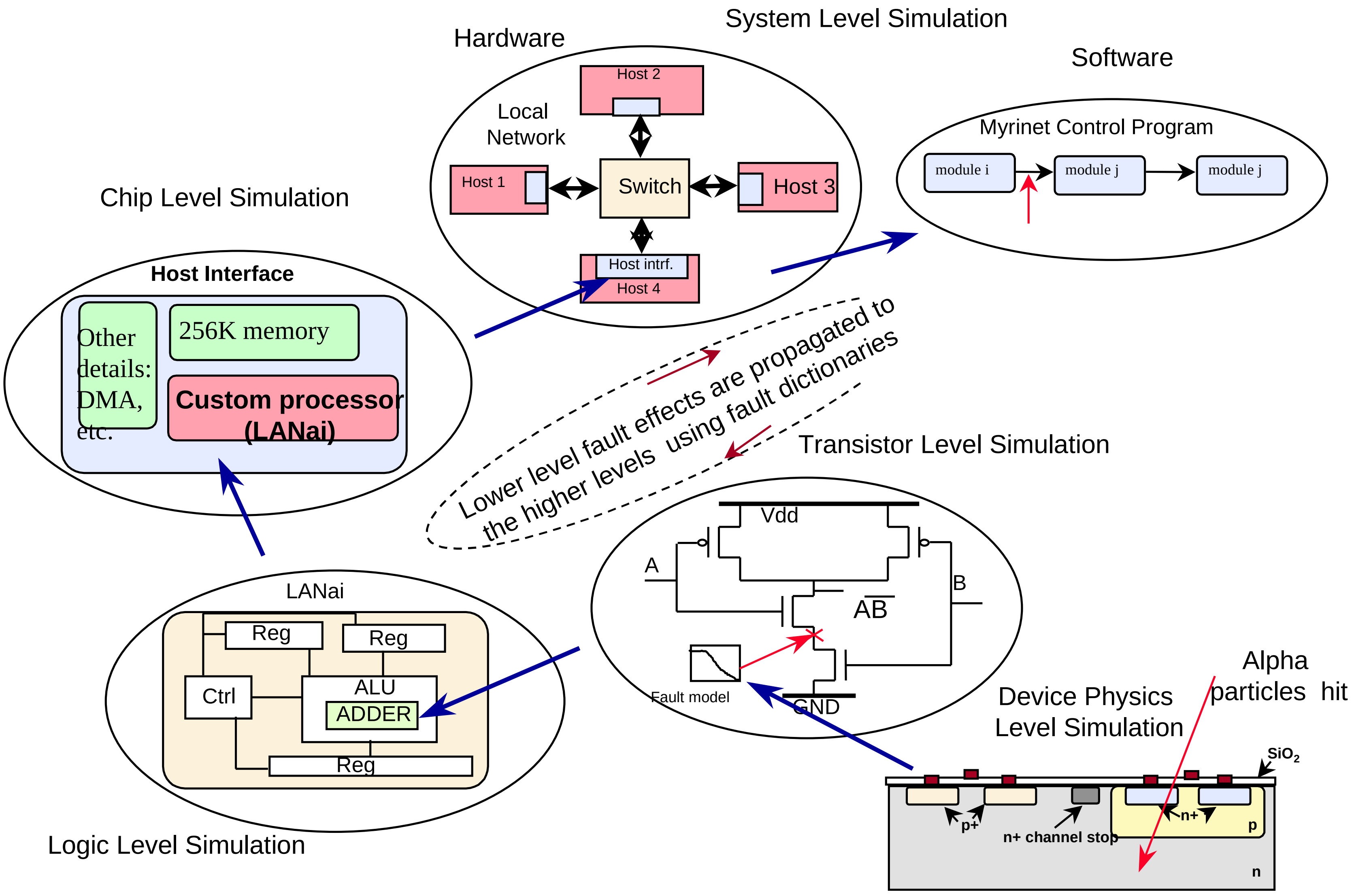
# Issues

- **Conventional methods of simulated fault injection employ random bit-flipping at lower levels (e.g., gate or register) and evaluate resulting effects at higher levels (e.g., subsystem or system).**

- **Not based on a physical fault model and may lead to an unrealistic predictions at the system level.**

- **Accurate results possible by:**
  - generating transient faults based on a physical model that depicts transistor-level effects of cosmic radiation, current and voltage spikes, and bridging transients
  - propagating the effects of derived transients through to the system level.

# Hierarchical Fault Modeling- LANai Processor



System Level Simulation

Hardware

Software

Chip Level Simulation

Transistor Level Simulation

Device Physics Level Simulation

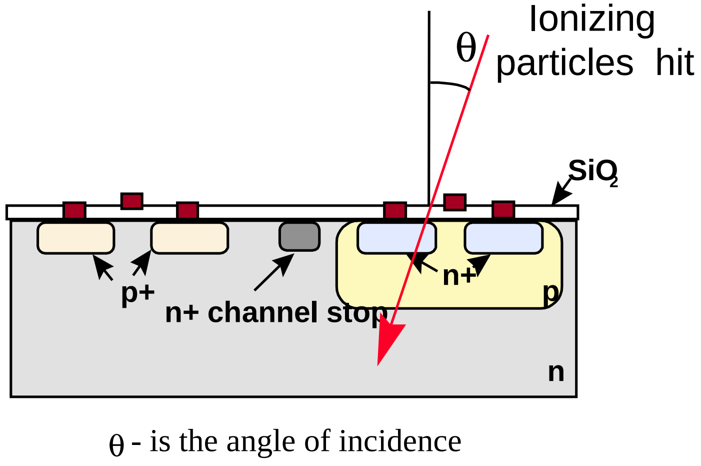
Logic Level Simulation

Lower level fault effects are propagated to the higher levels using fault dictionaries

Alpha particles hit

# Transistor Level Simulation

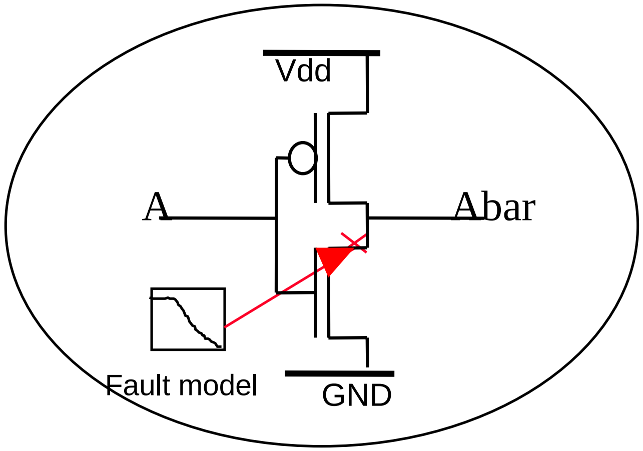
DESSIS Simulation



(a)

**(a) Ionizing Particle Incident on a CMOS Inverter,**

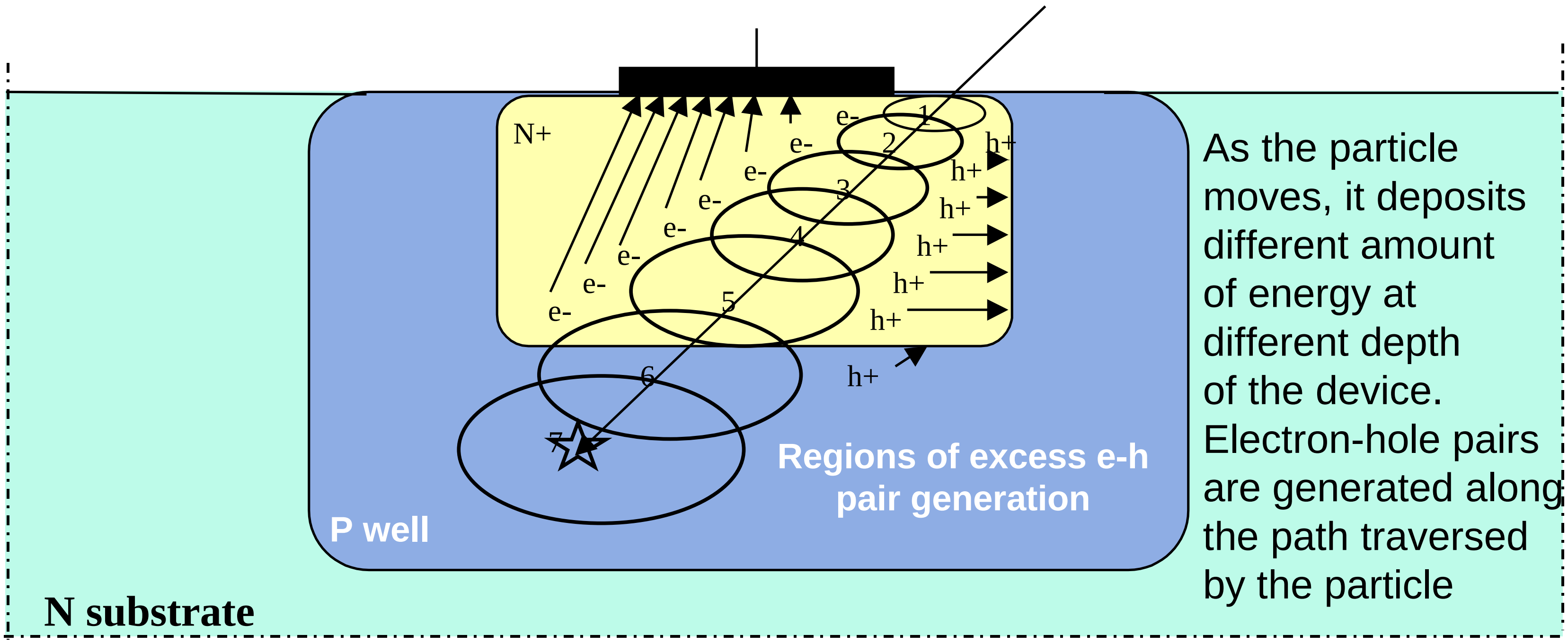
HSPICE Simulation



(b)

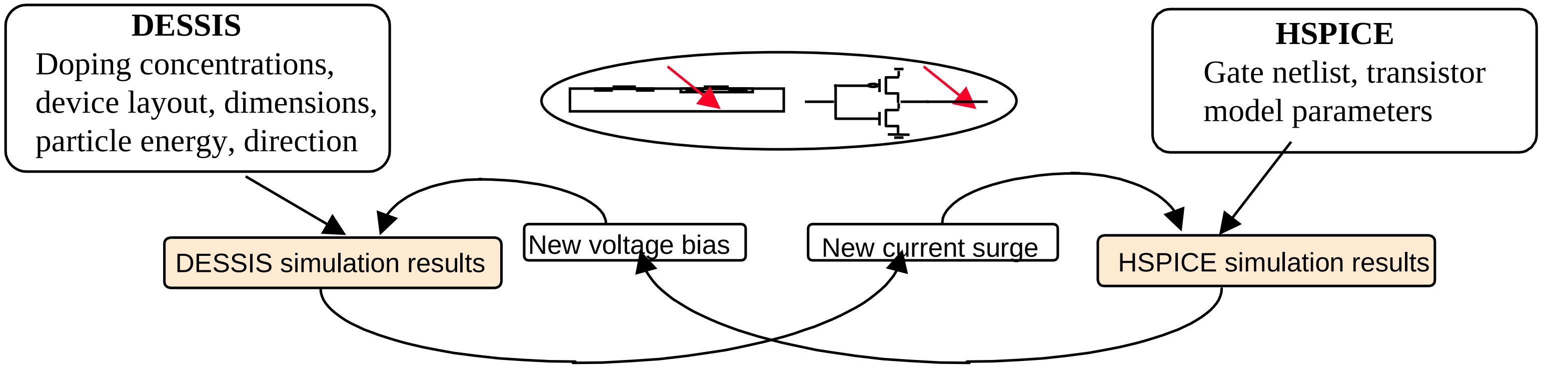
**(b) Inverter Circuit with Current Burst Injection at the Output Node**

# Injection Process in the Reverse Biased Junction of NMOS Transistor



Regions of excess e-h pair generation

As the particle moves, it deposits different amount of energy at different depth of the device. Electron-hole pairs are generated along the path traversed by the particle



**DESSIS**
Doping concentrations, device layout, dimensions, particle energy, direction

**HSPICE**
Gate netlist, transistor model parameters

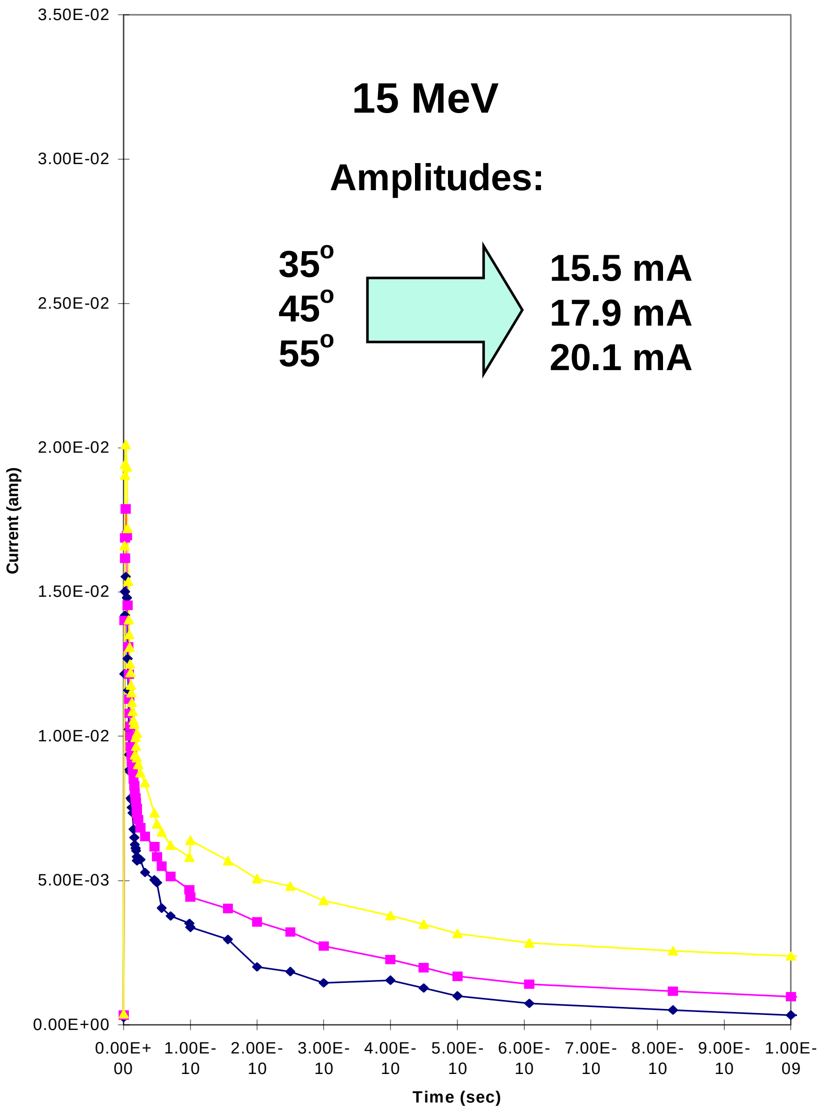
DESSIS simulation results

New voltage bias

New current surge

HSPICE simulation results

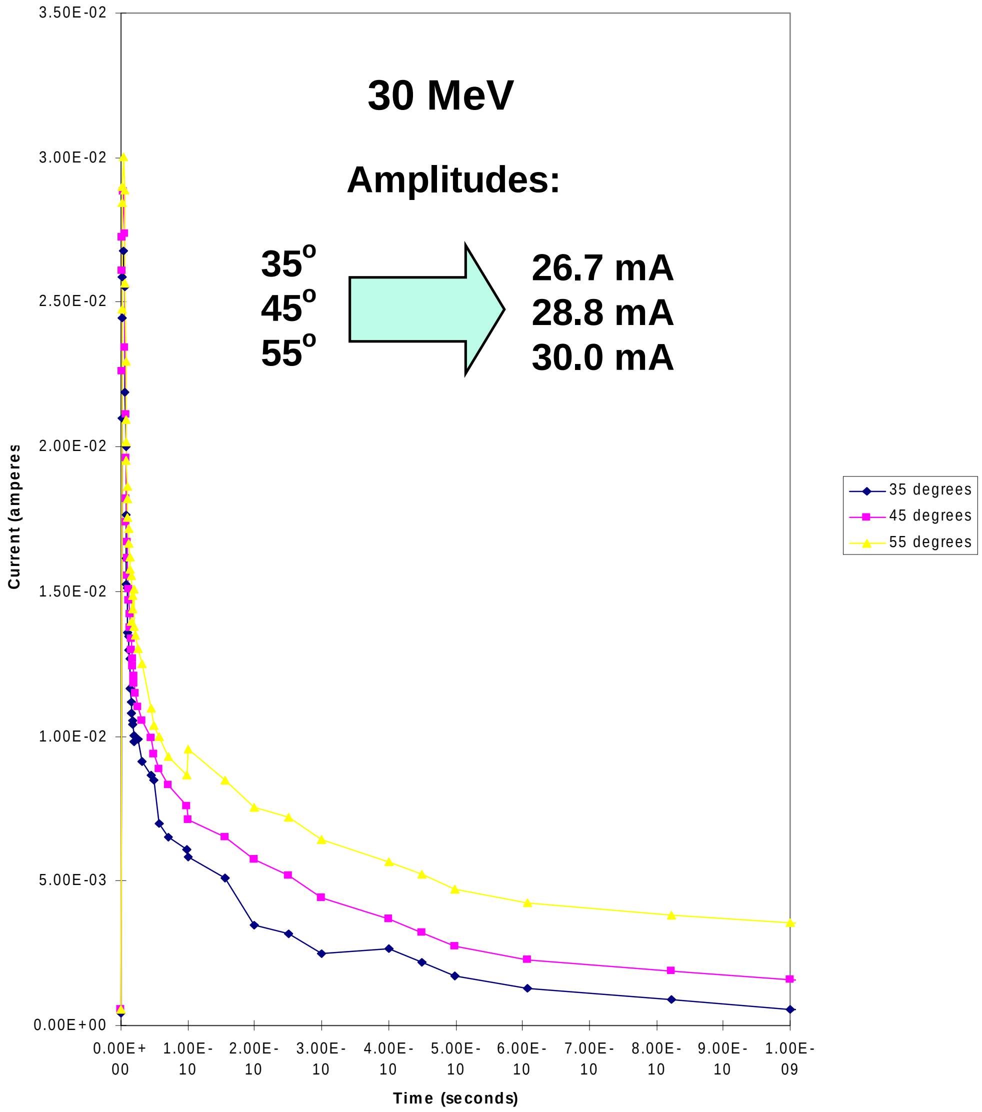
# Primary Fault Models Derived from Mg12+ Ion



15 MeV

Amplitudes:

$35^o$ → 15.5 mA
$45^o$ → 17.9 mA
$55^o$ → 20.1 mA



30 MeV

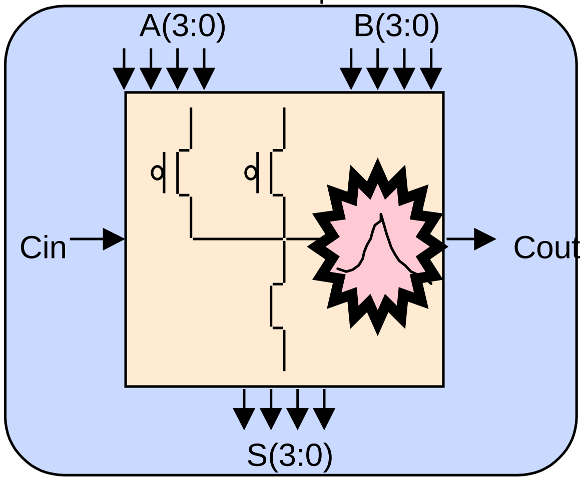
Amplitudes:

$35^o$ → 26.7 mA
$45^o$ → 28.8 mA
$55^o$ → 30.0 mA

# Circuit Level Simulation

Primary Fault Model

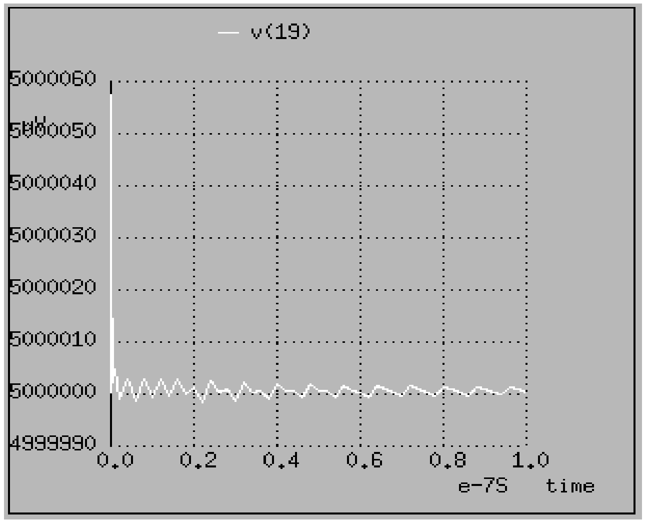
Transistor Level Description of 4-bit Adder



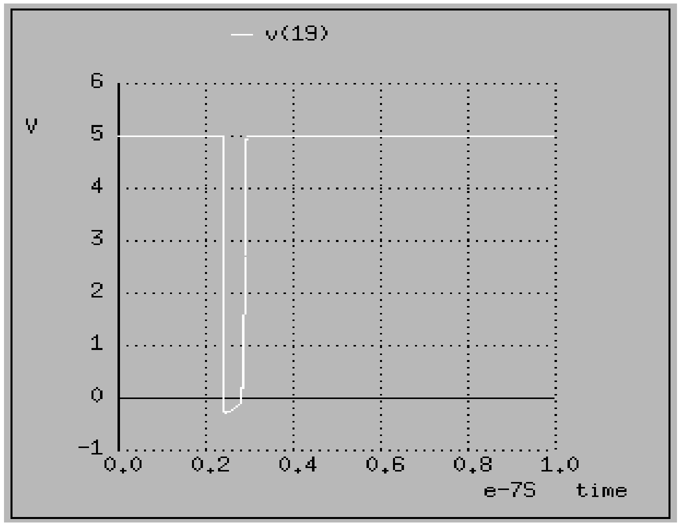
For all nodes, for all input combinations

Circuit Level Fault Dictionary

| | A | B | Cin |
|---|---|---|---|
| Input | 0000 | 0000 | 0 |
| | **S** | **Cout** | |
| | ---- | F | 34% |
| | ---F | F | 39% |
| | --F- | F | 7% |
| | F-F- | F | 20% |
| Input | 0000 | 0000 | 1 |
| | ---F | F | 10% |
| | -F-F | - | 2% |
| | -FF- | - | 52% |
| | -FF- | F | 13% |
| | F-FF | F | 23% |
| | : | | |
| | : | | |
| Input | 1111 | 1111 | 1 |
| | ---- | F | 23% |
| | ---F | - | 1% |
| | --FF | F | 9% |
| | -F-- | - | 33% |
| | -FFF | - | 33% |
| | FFFFF | | 1% |



Voltage at an inverter output without fault injection

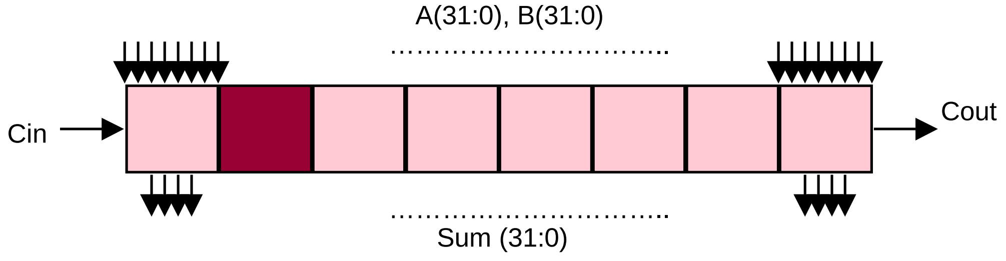


Voltage at an inverter output with fault injection
(latching at 25ns)

**University of Illinois - CRHC**

# Chip Level Simulation

Circuit Level Fault Dictionary → 32-bit Adder in LANai Chip → Chip Level Fault Dictionary



Execution of assembly code of the Myrinet simulator with fault injections

```
Begin
        :
        :
   add      %r14, %r13, %r14

   mov      00000001, %r9

   st       %r9, 00000000[%r14]
        :
        :
End
```

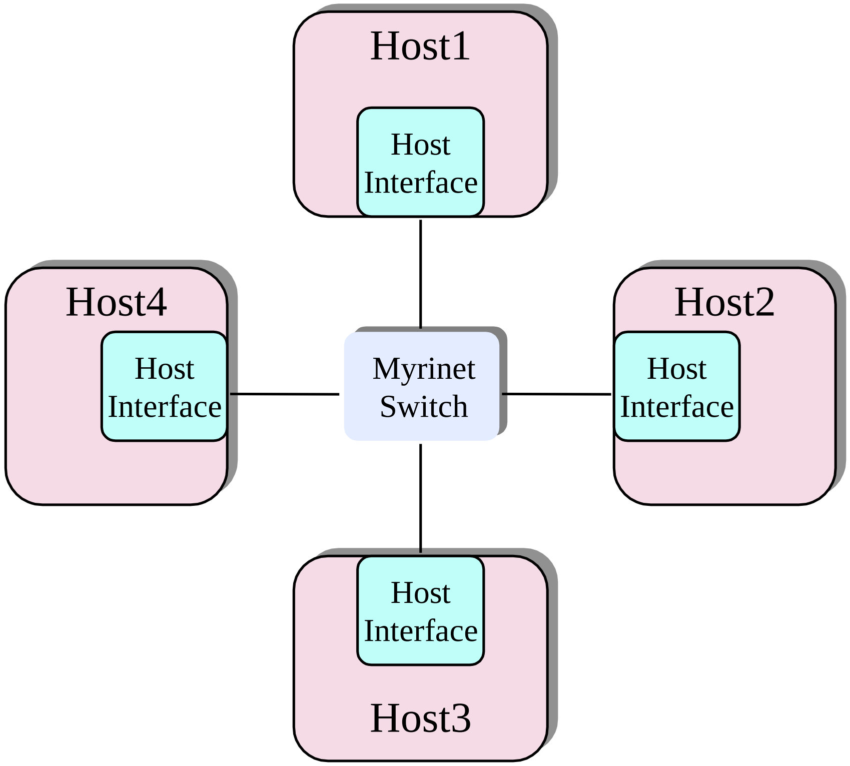
Comparison between correct and computed results

Faults injected while executing instructions that use adders.

```
Cycles:31
Run: 1
START
Memory Loc.  Corrupt   Correct
      13750     0          1
      13e04     10c6c   10ae8
      14b14     0          1
      14b18     13aa0      0
      14b22     13e10   13750

END

Cycles: 27
Run:  22
START
      13750       0          1
      13e04       10c6c 10ae8
      14b14       0          1
      14b18       13aa0      0

END
```

# System Level Simulation

Chip Level Fault Dictionary → Workstation Network Connected via Myrinet Switch → Normal/Abnormal System Behavior



| Fault Impact | 35 degrees | 45 degrees | 55 degrees |
|---|---|---|---|
| **Good** | 92 (56%) | 280 (56%) | 269 (55%) |
| **Hang** | 40 (25%) | 82 (16%) | 81 (17%) |
| **Corrupted** | 9 (6%) | 40 (8%) | 41 (8%) |
| **Dropped** | 22 (13%) | 98 (20%) | 97 (20%) |
| **Reset** | 0 (0%) | 0 (0%) | 0 (0%) |
| **Total** | 163 (100%) | 500 (100%) | 487 (100%) |

| Fault Impact | Without recovery | With recovery |
|---|---|---|
| **Good** | 280 | 327 |
| Hang | 82 | 22 |
| **Corrupted** | 40 | 40 |
| **Dropped** | 98 | 111 |
| **Reset** | 0 | 0 |
| **Total** | 500 | 500 |

# Hierarchical Simulation - Conclusions

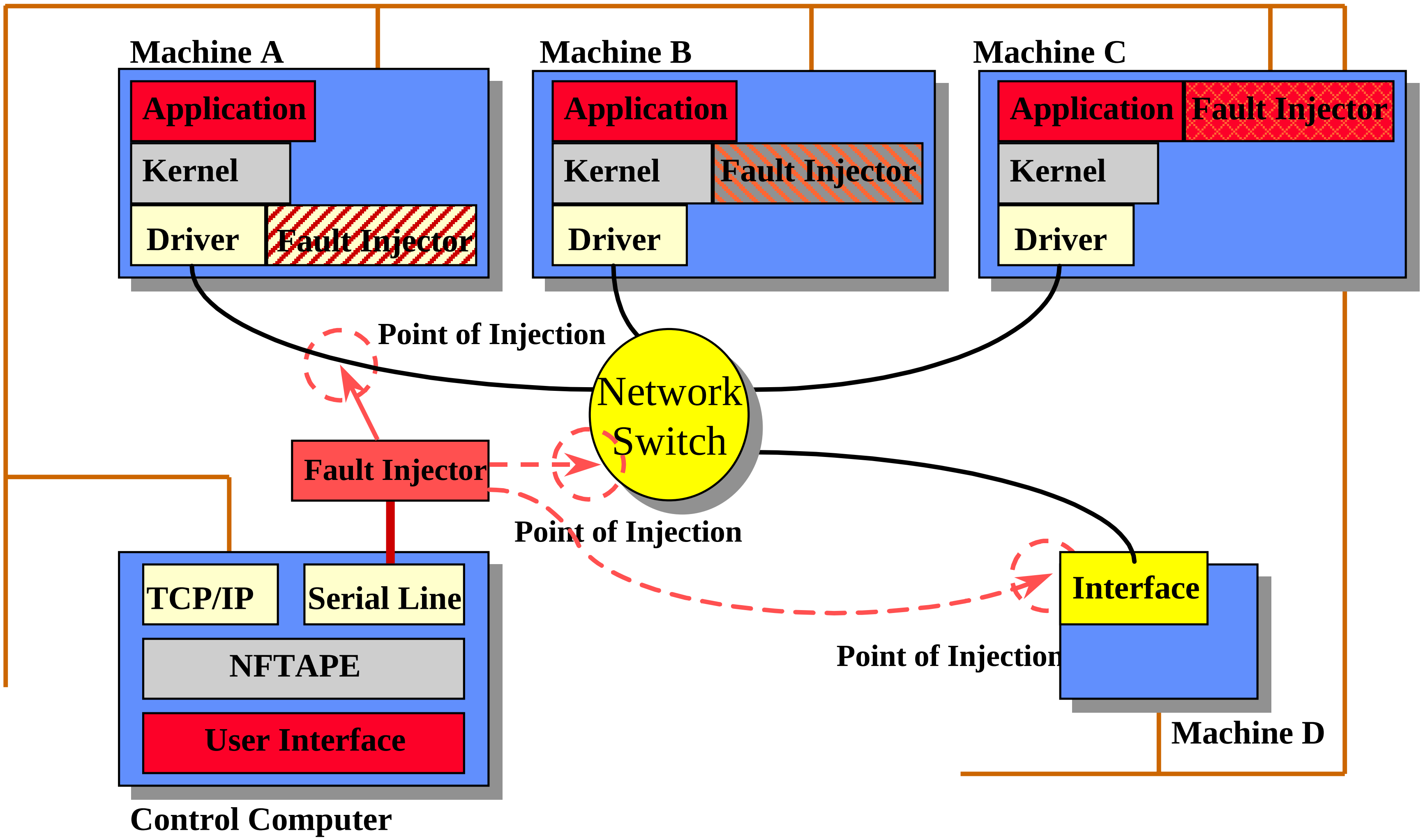
- **The hierarchical simulation approach offers:**
  - injection of representative faults
  - system evaluation on different abstraction levels (from device level to system level)
  - rapid propagation of low level transients through the system with realistic workloads
  - high confidence in the evaluation and validation results

- **The approach is being integrated to the DEPEND framework for designing and validation of dependable computing and communication systems.**

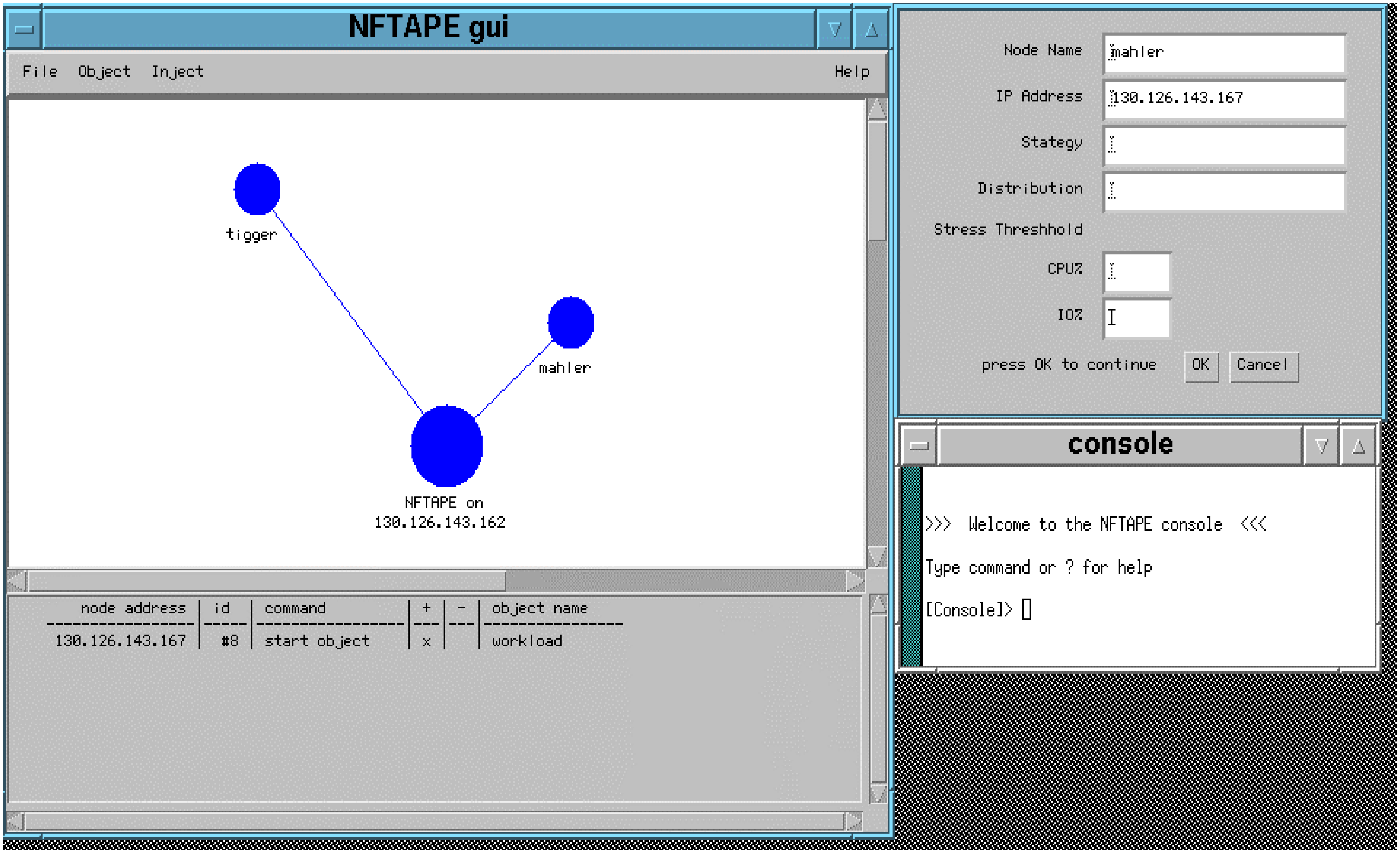
- **Web site: http://www.crhc.uiuc.edu/DEPEND**

# N-FTAPE: Network Fault Tolerance and Performance Evaluator

# N-FTAPE


Machine A
Application
Kernel
Driver
Fault Injector
Machine B
Application
Kernel
Fault Injector
Driver
Machine C
Application
Fault Injector
Kernel
Driver
Point of Injection
Network Switch
Fault Injector
Point of Injection
TCP/IP
Serial Line
NFTAPE
User Interface
Control Computer
Interface
Point of Injection
Machine D

# N-FTAPE User Interface



| node address | id | command | + | - | object name |
|---|---|---|---|---|---|
| 130.126.143.167 | #8 | start object | x | | workload |

Node Name: mahler

IP Address: 130.126.143.167

Stategy:

Distribution:

Stress Threshhold

CPU%:

IO%:

press OK to continue OK Cancel

```
>>>  Welcome to the NFTAPE console  <<<

Type command or ? for help

[Console]> 
```

# Validation of Simulation Using SWIFI

## Dependability study of Myrinet networked system

**Simulated Myrinet Network - simulated fault injection (DEPEND)**

**Real Myrinet Network - Software Implemented Fault Injection**

- **simulation does extremely well in modeling non faulty conditions (97%)**
- **reasonable well at predicting less severe injection results, e.g., a dropped message (95%)**
- **relatively low accuracy in predicting severe faults, e.g., interface hang (62%)**
- **Reasons:**
  - simplification of the simulation model, e.g., the interaction between the components in the system - a host and the interface
  - specification problems, e.g., not specified behavior for a response to an unaligned memory address
  - impact of the simulation environment, e.g., the simulation engine and the executed software shared the same process and an error in the simulated software can corrupt variables belonging to the simulation engine

# Summary & Conclusions

## Framework for Design and Evaluation of Dependable systems:

- Hierarchical Methodology for Accurate fault Modeling
- horizontal hierarchy
  - system decomposition into subsystems at the same abstraction level
  - demonstrated - cache-based RAID storage systems
- vertical hierarchy
  - system decomposition into subsystems at different abstraction levels, e.g., device level, circuit level, chip level and system level.
  - demonstrated - Myrinet network system
- effects of low-level faults are propagated to the higher levels using fault dictionaries
- developed library of components to simulate systems with different characteristics, e.g., different network systems - Myrinet, ServerNet (by Tandem)

## Network Fault Tolerance and Performance Evaluator

### Software Implemented Fault Injection

- non-intrusive method of introducing faults to:
  - memory locations, CPU registers, and I/O (including network) to affect application and/or operating system
- injection under a realistic workload (real programs or synthetic workload)
- ability to mimic real fault conditions
- demonstrated:
  - Integrity S2 fault -tolerant system (by Tandem)
  - Sun systems Sun OS, Sun Solaris (on sun4m), Sun Solaris (on sun4u ongoing)
  - Windows NT - study of fault impacts in device drivers - ServerNet network,

### *Hardware Implemented Fault Injection*

- *non-intrusive injection into signals transmitted through the network*
- *high speed of operation (system clock from 500Mhz to several GHz)*
- *triggers and injection patterns stored in FPGA*
- *the manufactured prototype is under test*

# Collaboration with JPL

**Methods and tools to design and evaluate systems with enough detail and accuracy to reveal design flows and assess system dependability early in the design process.**

## Early design phase

### Framework for Design and Evaluation of Dependable systems

- CAD environment to evaluate design via simulation
- hierarchical methodology for accurate fault modeling
- simulated fault injection experiments
- evaluate effectiveness of fault tolerance mechanisms (e.g., coverage, detection latency)

## Work with JPL

- we collaborate with JPL in using DEPEND framework for conducting experiments with low level fault injection and N-FTAPE for supporting fault injection to actual systems
    - DEPEND will be ported soon to JPL
    - N-FTAPE is being ported to JPL
- our students will work remotely and in JPL site to ensure correct operation of the tools (currently two students are working with JPL)
- target systems will be identified with close collaboration with the JPL scientists and engineers (e.g., a VxWorks based networked system)

## Prototype phase

### Network Fault Tolerance and Performance Evaluator

### Software Implemented Fault Injection

- non-intrusive method of introducing faults
- injection under a realistic workload
- measures: error latency and propagation times, detection distributions

## JPL should provide

- computing resources, including hardware (e.g., computing platforms) and software (e.g., applications and development tools)
- details about selected computing platforms, including configuration of computation nodes, type of operating system, I/O, network
- contact person(s) familiar with specifics of JPL environment and capable of providing necessary technical assistance